Parul Tyagi, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Comparison Study on Different Neural Network Techniques for Kidney Stone Diagnosis

Parul Tyagi, Dr. Brij Mohan Singh
Department of Computer Science, Quantum University, Roorkee, UK

Abstract- Kidney stone disease — clinically referred to as nephrolithiasis — remains one of the most painful and widely encountered urological conditions worldwide. Catching it early and getting the diagnosis right can dramatically change a patient's care pathway and reduce the financial strain on healthcare systems. In this study, we take a close, side-by-side look at five machine-learning techniques that have shown promise for automated kidney-stone detection: the Multilayer Perceptron trained with Back Propagation (MLP-BPA), Radial Basis Function (RBF) networks, Learning Vector Quantization (LVQ), Support Vector Machines (SVM), and Deep Convolutional Neural Networks (CNN). All experiments are run on a standardised clinical dataset using WEKA 3.7.5 and Python, with each model assessed on accuracy, sensitivity, specificity, and F1-score. The features fed into every model include creatinine and BUN levels, CT-scan findings, kidney size and contour, and several urinary markers. Among the classical approaches, SVM came out on top with 93.6% accuracy, while MLP-BPA was close behind at — a CNN — pushed accuracy to 96.1% when adequate training images were available. Beyond the raw numbers, we discuss what each architecture actually trades off in practice: how hard it is to train, how transparently it reaches its decisions, and whether a busy nephrology clinic could realistically deploy it. Our hope is that this comparison gives clinicians and AI researchers a clear, honest basis for choosing the right tool for kidney stone diagnosis.

Keywords— Kidney stone diagnosis, Artificial Neural Network, MLP, RBF, LVQ, SVM, Deep Learning, WEKA, Medical classification, Nephrolithiasis

I. INTRODUCTION

There is a reason doctors have traditionally been described as practising medicine — it takes years of accumulated experience, a sharp eye for subtle patterns, and a willingness to hold multiple competing hypotheses in mind at once. A good clinician listens to the patient, reviews the numbers, studies the images, and synthesises it all into something actionable. That process has always been hard, and it is getting harder: the volume of data generated during a single patient visit continues to grow, and expecting a physician to process every data point without any assistance is no longer realistic.

Kidney stone disease sits right at the centre of this challenge. Stones form when minerals — mainly calcium and oxalate, and sometimes uric acid — build up in the urine faster than they can be diluted and flushed out. They start as microscopic crystals and can grow over months into obstructions that wedge themselves into the ureter, triggering the kind of pain patients routinely describe as worse than childbirth. The condition is far from rare: somewhere between ten and fifteen percent of people in high-income countries will pass a kidney stone at some point in their lives, and more than half of those will face a recurrence within a decade. The costs — hospital stays, procedures, lost working days — amount to billions of dollars every year across the globe.

What makes kidney stone diagnosis tricky is that it does not rest on a single test. A clinician typically has to weigh CT-scan results, renal ultrasound findings, blood chemistry (creatinine, BUN), urine composition, and the patient's own history of symptoms before committing to a diagnosis. Pulling

all of that together reliably, especially in a busy outpatient setting or a rural clinic without a specialist radiologist, is genuinely difficult. That is where machine learning, and specifically artificial neural networks, can add real value.

Neural networks are well suited to this kind of problem. They do not need an expert to hand-craft rules about which combinations of features point to a stone — instead, they learn those relationships directly from labelled patient records. They can handle the messy, non-linear interactions between variables like kidney size, urinary flow rate, and creatinine levels without those interactions having to be spelled out explicitly. And once trained, they can return a classification in milliseconds, which matters enormously in high-throughput clinical workflows.

This paper pits five well-established approaches against each other on a standardised kidney-stone dataset: the classic Multilayer Perceptron with Back Propagation (MLP-BPA), Radial Basis Function networks (RBF), Learning Vector Quantization (LVQ), Support Vector Machines (SVM), and Deep Convolutional Neural Networks (CNN). We report accuracy, sensitivity, and F1-score for each, but we also try to go beyond the numbers and discuss what each model is actually like to work with — how long it takes to train, how easy it is to explain to a clinician, and where its practical limits lie.

Background and Related Work

Kidney Stone Disease: Pathophysiology and Epidemiology

Not all kidney stones are alike. The vast majority — roughly eighty percent — are composed of calcium oxalate, and these tend to be the most clinically problematic. Uric acid stones make up somewhere between five and ten percent of cases, often linked to gout or a diet heavy in red meat and shellfish. Struvite stones are a product of infection: bacteria that produce urease cause ammonia to accumulate in the urine, which raises its pH and allows magnesium ammonium phosphate crystals to precipitate. Finally, cystine stones are comparatively rare, arising from an inherited disorder that prevents the kidneys from reabsorbing the amino acid cystine. Each stone type has its own risk profile and, ideally, its own management strategy — which is one reason accurate classification matters.

Geography plays a surprisingly large role in who gets kidney stones. Incidence is highest across Scandinavia, the Mediterranean basin, the British Isles, northern Australia, central Europe, and parts of South Asia, particularly Pakistan and northern India. Rates tend to be lower in Central and South America and across much of sub-Saharan Africa, though the underlying reasons — climate, diet, hydration habits, access to healthcare — are still being untangled. What is clear is that the disease cuts across age groups, affecting patients from infancy right through to their seventies and beyond. Men are historically about three times more likely to develop kidney stones than women, though that gap has been narrowing over recent decades as female obesity and dietary patterns have shifted closer to male norms. Of all the modifiable risk factors — and there are many — diet stands out most clearly: high animal protein intake, excessive sodium, and chronically low fluid consumption are consistently implicated across populations.

Key Diagnostic Features

Building any automated diagnostic system for kidney stones requires settling on a set of input features that are both clinically meaningful and routinely collected. After reviewing the literature and consulting the dataset used in this study, we identified the following as the most informative variables:

- Patient demographics: sex, age, and body mass index
- Kidney morphology: which side is affected (left, right, or bilateral), size category (very small, small, normal, or enlarged), and contour shape (normal, irregular, lobulated, or overtly abnormal)
- Imaging findings: CT-scan interpretation (negative, mass, stone, cyst, or abscess) and ultrasound echogenicity where available
- Biochemical markers: serum creatinine, blood urea nitrogen (BUN), and white cell count
- Urinalysis: urine volume category (very low through high), flow pattern (weak, normal, intermittent, or dribbling), and specific abnormal

findings — haematuria, proteinuria, bacteriuria, elevated white blood cells, or elevated red blood cells

Neural Networks in Medical Diagnosis — A Brief Survey

The idea of using neural networks to support clinical decision-making is not new. Work stretching back to the early 1990s showed that relatively simple multilayer networks could match or outperform logistic regression on tasks like diagnosing myocardial infarction, staging prostate cancer, interpreting ECG rhythms, and classifying STDs. By the late 1990s, neural network tools were being applied to NMR and PET scan analysis, cytological Pap smear screening, WBC differential counting, and surgical outcome prediction — a breadth of application that few other computational methods have matched.

The field gathered considerable momentum in the decade that followed. Catalogna et al. (2012) showed that ANNs could usefully analyse the blood and urine signatures of diabetic patients, while Er et al. (2008) and Elveren and Yumuşak (2011) demonstrated their value in tuberculosis diagnosis from routine clinical parameters. Dey et al. (2012) applied them to leukaemia classification, and Barbosa et al. (2012) and Saghiri et al. (2012) pushed into radiographic image analysis of living tissue. Each of these studies reinforced the same basic point: neural networks are remarkably versatile, and their performance tends to improve as the quality and quantity of training data increases. The arrival of deep learning after AlexNet's breakthrough in 2012 brought a step change in image-based diagnostic accuracy, and convolutional architectures now underpin many of the most capable radiology AI tools in clinical use today.

II. METHODOLOGY

1. Dataset Description

The dataset used in this study contains 400 patient records drawn from the outpatient nephrology department of a regional hospital. Every record includes 24 attributes — covering all the diagnostic features described in Section 2.2 — along with a binary label indicating whether a kidney stone was confirmed (Stone Present) or not (Stone Absent). Before any modelling began, we spent time cleaning the data: missing values in continuous attributes were replaced with column means, while missing categorical values were filled with the mode for that column. All continuous features were then scaled to the [0, 1] range so that no single variable could dominate the learning process simply because it was measured on a larger numerical scale. To get reliable performance estimates without wasting data on a separate holdout set, we used stratified 10-fold cross-validation throughout all experiments, ensuring that each fold preserved the original class balance.

2. Neural Network Models

Multilayer Perceptron with Back Propagation (MLP-BPA)

The MLP is about as foundational as neural networks get. At its core, it is a stack of fully connected layers: an input layer that takes in the 24 clinical features, one or more hidden layers that apply non-linear transformations (using sigmoid or ReLU activation functions), and an output layer that emits a class prediction. What makes it learn is back-propagation — after each forward pass, the gap between the predicted output and the true label is used to compute gradients, which then flow backwards through the network and nudge every weight slightly in the direction that reduces the error. We ran the network for 500 epochs with a learning rate of 0.3 and a momentum term of 0.2 to help push through flat regions of the loss surface. The MLP's main appeal is its generality — given enough hidden units and training data, it can approximate virtually any continuous function. Its main frustration is that getting it to converge reliably can involve a lot of trial and error with the hyperparameters.

Radial Basis Function (RBF) Network

RBF networks take a rather different approach to the same problem. Instead of learning a global representation spread across many layers, each hidden unit in an RBF network responds to how close an input is to a specific learned reference point, called a centre. The response follows a Gaussian curve: $\varphi_j(x) = \exp(-\|x - c_j\|^2 / 2\sigma_j^2)$, where c_j is

the centre and σ_j controls how quickly the response falls off with distance. In practice, the centres are found by running k-means clustering on the training data — an unsupervised step that does not require labels — and then the output-layer weights are fitted by straightforward linear regression. This two-phase training is much faster than gradient descent over the whole network, which makes RBF networks attractive when turnaround time matters. They tend to perform best when the data naturally forms tight, well-separated clusters in feature space, which is reasonably likely given the distinct clinical profiles of different kidney-stone types.

Learning Vector Quantization (LVQ)

LVQ operates on a deceptively simple idea: maintain a small set of reference vectors — one or a few per class — and update them based on whether they get the label right. When a training example arrives, the algorithm finds the closest reference vector. If that vector carries the correct class label, it gets nudged a little closer to the example (a reward step). If it has the wrong label, it gets pushed further away (a penalty step). Over time, the reference vectors settle into positions that best summarise their respective classes, and classification of a new example is simply a matter of assigning it the label of whichever reference vector it falls nearest to. The resulting model is highly transparent — the reference vectors themselves describe what a typical stone patient or a typical stone-free patient looks like in feature space — which is a genuine advantage in medical contexts where clinicians want to understand why a system reached its conclusion.

Support Vector Machine (SVM)

The SVM approaches classification as a geometry problem: find the hyperplane that divides the two classes with the largest possible margin — that is, the greatest possible distance between the plane and the nearest training examples on either side. For a dataset like ours, where the classes are not linearly separable in the original 24-dimensional feature space, the trick is to implicitly map the data into a higher-dimensional space using a kernel function. We used the RBF kernel $K(x, x') = \exp(-\gamma||x - x'||^2)$, which has proven effective on many clinical datasets. The optimisation is solved using the Sequential Minimal Optimisation algorithm, and the final model depends only on the training examples that end up on or near the margin — the support vectors. This sparsity makes inference fast and the model relatively resistant to noise in the training data, though it does mean that choosing the regularisation constant C and the kernel bandwidth γ requires careful cross-validated tuning.

Deep Convolutional Neural Network (CNN)

CNNs earned their reputation in computer vision, but they are increasingly finding application in structured medical data as well. When CT-scan images were available in our dataset, we fed them through a five-layer convolutional architecture: three convolutional blocks, each consisting of a convolutional layer, batch normalisation, ReLU activation, and max-pooling, followed by two fully connected layers with dropout ($p = 0.5$) applied between them to reduce overfitting. For records where only tabular clinical features were available, we used a one-dimensional variant of the CNN that applies filters along the feature axis rather than across spatial dimensions. The CNN takes noticeably longer to train than the other models — a GPU is more or less essential for any reasonable turnaround — and it needs more data to generalise reliably. But when those conditions are met, its ability to discover subtle, hierarchical patterns in the input that a human annotator might never think to encode explicitly gives it a clear edge in raw accuracy.

3. Experimental Setup — WEKA 3.7.5

For the three classical algorithms (MLP-BPA, RBF, and LVQ), we used WEKA 3.7.5 as the primary experimental platform. WEKA was a natural choice: it packages well-tested implementations of all three classifiers, handles cross-validation bookkeeping automatically, and produces a consistent set of evaluation metrics that makes side-by-side comparison straightforward. Table 1 details the specific WEKA classifier names and the parameter values we settled on after initial tuning runs.

Table 1: WEKA Configuration for Classical Neural Network Models

Algorithm	WEKA Classifier	Key Parameters
MLP / BPA	Multilayer Perceptron (functions. Multilayer Perceptron)	Learning rate 0.3, Momentum 0.2, 500 epochs
RBF Network	RBF Network (classifiers. functions. RBF Network)	Num clusters = 2, Ridge = 1.0E-8
LVQ	VFI / LVQ custom implementation	Num prototypes = 5 per class
SVM	SMO (functions. SMO) with RBF kernel	C = 1.0, γ = 0.01

IV. RESULTS AND DISCUSSION

1. Comparative Performance

Table 2 lays out a broad architectural comparison across all five models, while Table 3 reports the classification metrics that came out of 10-fold cross-validation. Looking at the numbers together makes it easier to see not just who won, but why, and what that means for real-world use.

Table 2: Comparative Analysis of Neural Network Techniques

Model / Technique	Learning Type	Core Mechanism	Complexity	Training Speed	Accuracy (%)
MLP (Back Propagation)	Supervised	Gradient descent weight update	High (multiple hidden layers)	Slow convergence	91.3%
Radial Basis Function (RBF)	Supervised	Gaussian basis functions	Moderate	Fast training	88.7%
Learning Vector Quantization (LVQ)	Supervised	Prototype-based competitive learning	Moderate	Moderate	85.4%
Support Vector Machine (SVM)	Supervised	Kernel-based margin maximization	Low–Moderate	Moderate	93.6%
Deep CNN	Supervised	Hierarchical feature learning	Very High	Slow (GPU req.)	96.1%

Table 3: Classification Performance Metrics (10-fold Cross-Validation)

Model	Accuracy	Sensitivity	F1-Score	Remarks
MLP (BPA)	91.3%	89.5%	90.4%	Moderate overfitting risk
RBF Network	88.7%	87.1%	88.0%	Fast, good generalisation
LVQ	85.4%	84.0%	84.7%	Best interpretability
SVM (RBF kernel)	93.6%	92.8%	93.2%	High accuracy, sparse model
Deep CNN	96.1%	95.4%	95.7%	Best accuracy, needs large data

2. Discussion

MLP-BPA

With 91.3% accuracy and an F1-score of 90.4%, the MLP-BPA sits comfortably in second place among the classical models. It handled the non-linear relationships between variables well — which makes sense, given that back-propagation was specifically designed to navigate exactly those kinds of complex feature interactions. That said, getting there required patience. Training the network to the point where it stopped improving took a fair number of epochs, and the results were sensitive to small changes in the learning rate and momentum settings. For a kidney-stone dataset where the interactions between CT findings, creatinine levels, and urinary abnormalities are genuinely tangled, the MLP's representational power paid off. In simpler domains, the overhead might not be worth it.

RBF Network

The RBF network landed at 88.7% accuracy. That puts it a step behind the MLP-BPA on pure performance, but the trade-off it offers is genuinely appealing: training was substantially faster, because the centre-finding phase relies on k-means rather than iterative gradient descent. In a clinical setting where patient cohorts shift over time and models need to be updated regularly, that speed advantage is not trivial. The architecture is also well-matched to data with distinct sub-populations — and given that calcium oxalate, uric acid, and struvite stones have meaningfully different clinical profiles, the kidney-stone dataset seems like a reasonable fit. Where the RBF struggled was at the edges of its clusters, where ambiguous cases did not sit neatly close to any single centre. Increasing the number of centres, or adapting them more aggressively during training, could address that.

LVQ

At 85.4% accuracy, LVQ finished last among the five models — but we think that headline figure undersells what it brings to the table. Its learned prototype vectors are not just mathematical abstractions; they represent something a clinician can actually look at and reason about. Prototype 1 might describe a typical stone-positive patient: male, middle-aged, elevated creatinine, small irregular kidney, CT showing a stone. Prototype 2 might be stone-negative: female, young, normal biochemistry, clean CT. That kind of direct inspectability is rare in machine learning, and in safety-critical medical applications it can make the difference between a tool that gets adopted and one that sits unused because the clinical team does not trust what they cannot understand. Accuracy could likely be improved by using more prototypes per class, or by switching to higher-order variants like LVQ2.1 or LVQ3 that specifically sharpen the decision boundary.

SVM

The SVM was the strongest classical model at 93.6% accuracy, and this result was fairly robust across different folds — the standard deviation across folds was low, which suggests the model was not just getting lucky on a few easy partitions. The RBF kernel allowed it to capture the non-linear structure of the data effectively, and the maximum-margin criterion gave it a degree of built-in resistance to the occasional noisy or mislabelled record. Inference is also fast at deployment time, since the model only needs to evaluate the kernel function against the support vectors rather than the full training set. The main gotcha is hyperparameter sensitivity: both C and γ needed careful tuning via grid search, and poorly chosen values could push accuracy down by several percentage points.

Deep CNN

The CNN reached 96.1% accuracy — the highest of any model we tested — with a sensitivity of 95.4% and an F1-score of 95.7%. When CT-scan images were incorporated, the convolutional layers picked up on textural and morphological features that would be extremely difficult to encode as hand-crafted inputs: the precise density gradients inside the kidney parenchyma, the subtle thickening of the urothelial wall, the shadowing artefacts that appear around calcified stones on ultrasound. Those features are invisible to a tabular model that receives only the radiologist's categorical summary of the scan. The price for this performance is real, though. The CNN was the most computationally demanding model by a wide margin, and it showed clear signs of overfitting on smaller training subsets — which is

why we leaned heavily on dropout and why data augmentation would likely be necessary in a production setting with a more limited patient registry.

Applications of Neural Networks in Medical Diagnosis

The techniques evaluated in this paper are not limited to kidney stone detection. The same fundamental architectures — MLP, RBF, LVQ, SVM, CNN — have been applied productively across a wide range of clinical domains. Table 4 gives an overview of the most established application areas, which collectively illustrate both how versatile these tools are and how the choice of architecture tends to follow the structure of the problem at hand.

Application Domain	Representative Tasks	Key Neural Network Type
Oncology	Prostate, lung, and colorectal carcinoma staging; tumour boundary segmentation	CNN, MLP
Cardiology	Arrhythmia classification, ECG interpretation, heart failure prognosis	LSTM, MLP
Radiology & Imaging	X-ray, NMR, PET, and perfusion-scan interpretation	Deep CNN
Haematology	WBC differential counting, leukaemia subtype classification	RBF, CNN
Infectious Disease	Tuberculosis, STD, and COVID-19 severity classification	MLP, SVM
Diabetology	Blood and urine sample analysis, insulin dosing prediction	RBF, LSTM
Neurology	Dementia grading, EEG seizure detection	LSTM, CNN
Surgery	Post-operative outcome prediction, complication risk stratification	MLP, SVM

A pattern worth noting across all these domains is that deep learning architectures (CNN, LSTM) tend to dominate wherever the raw input is high-dimensional and structured — images, waveforms, time-series physiological data — while classical methods like SVM and MLP remain competitive, and often preferable, when the input is tabular and the training dataset is modest in size. For clinical teams considering which approach to adopt, that distinction is probably the single most useful heuristic.

V. CONCLUSION

We set out to answer a practical question: among five well-known neural network techniques, which one gives a clinician the best shot at accurately identifying kidney stone disease from routine clinical and biochemical data? Having run all five on the same dataset under the same experimental conditions, here is what we found:

- The SVM achieved 93.6% accuracy — the best among the classical machine-learning models — and is our recommended choice for settings where labelled training data is limited and computational infrastructure is constrained. Its speed at inference time and its robustness to outliers are additional marks in its favour.
- The Deep CNN pushed accuracy to 96.1% when CT-scan images were available, making it the strongest overall performer. It is the right choice when imaging data is plentiful and a GPU-equipped server is available for training. For smaller clinics without that infrastructure, its overhead may outweigh its accuracy advantage.
- MLP-BPA (91.3%) is a solid general-purpose baseline. It handles non-linear interactions well

and is available in essentially every machine-learning toolkit, but it demands careful hyperparameter tuning and is slower to train than the alternatives.

- RBF networks (88.7%) offer the best training speed among the models we tested. In environments where the model needs to be regularly retrained on incoming patient data, that efficiency makes them a genuinely practical option, even at a modest cost in peak accuracy.
- LVQ (85.4%) had the lowest accuracy but the highest interpretability. For clinical teams that need to understand and explain the model's reasoning — in regulatory submissions, audit processes, or patient conversations — LVQ's prototype-based representation is a real advantage that the numbers alone do not capture.

Looking ahead, we see several promising directions. Ensemble methods that combine SVM predictions with MLP-BPA feature representations could push accuracy beyond what either achieves alone. Incorporating longitudinal patient data through recurrent architectures like LSTMs would allow models to track disease progression rather than making a single snapshot diagnosis. And explainability tools — SHAP values, attention maps, LIME — applied to the higher-accuracy models could help bridge the trust gap that currently limits their clinical uptake. Our longer-term goal is a deployable, transparent decision-support system that works alongside nephrologists rather than replacing them, helping them catch kidney stones earlier and with greater confidence.

REFERENCES

1. Robertson, W.G. et al. (1982). "Epidemiology of Urinary Stone Disease." Urological Research, 10(3), 141–154.
2. Evan, A.P. & Coe, F.L. (2007). "Kidney Stone Disease." Journal of Clinical Investigation, 115(10), 2598–2608.
3. Scales, C.D. et al. (2012). "Prevalence of Kidney Stones in the United States." European Urology, 62(1), 160–165.
4. Catalogna, M. et al. (2012). "Artificial Neural Networks in Blood Analysis of Diabetic Patients." Biomedical Engineering Letters, 2, 211–218.
5. Er, O. et al. (2008). "Tuberculosis Disease Diagnosis Using ANN." Expert Systems with Applications, 34(4), 2387–2394.
6. Dey, P. et al. (2012). "ANN for Leukemia Classification." Analytical Quantitative Cytology, 34, 229–234.
7. Barbosa, D.B. et al. (2012). "Image Analysis of Radiographs Using ANN." Dentomaxillofacial Radiology, 41, 297–304.
8. Saghiri, M.A. et al. (2012). "Application of Artificial Neural Networks in Endodontic Radiograph Analysis." Journal of Endodontics, 38(4), 478–482.
9. Elveren, E. & Yumuşak, N. (2011). "Tuberculosis Disease Diagnosis Using Artificial Neural Network Trained with Genetic Algorithm." Journal of Medical Systems, 35(3), 329–332.
10. Witten, I.H. & Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. Morgan Kaufmann.
11. Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, 2nd ed. Prentice Hall.
12. Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer.
13. Kohonen, T. (1990). "The Self-Organizing Map." Proceedings of the IEEE, 78(9), 1464–1480.
14. Broomhead, D.S. & Lowe, D. (1988). "Multivariable Functional Interpolation and Adaptive Networks." Complex Systems, 2, 321–355.